In the Specification:

Please amend the paragraph on page 14 beginning on line 13 as follows:

In one embodiment of the invention, the rule section 407 may include two types of rules: applicability rules 403 ~~407~~ and condition rules 409. The one or more rules in the rule section 407 may be formatted in a rule language such as but not limited to Knowledge Predicate Language (KPL). KPL may be formatted as "(predicate operand operand …)" where a predicate may be a functional statement and each operand may be one or more facts to fill a specific argument needed to evaluate the functional statement. For example, if an operand named "var1" is equal to 5 and another operand named "var2" is equal to 2, then a KPL statement "(set ?var3 (add ?var1 ?var2))" may set an operand named "var3" equal to 7. The predicate "add" may perform the function of adding the operands. Other predicates with predetermined functions may also be with in the scope of the invention.

Please amend the paragraph on page 17 beginning on line 1 as follows:

In the rule section 407, the one or more applicability rules 403 may be formatted according to rule language to use to evaluate whether the check is related to relevant product characteristics. For example, if the check is designed to detect product issues for an older version of software than is currently installed on the client's product, the one or more applicability rules 403 ~~407~~ may detect the different software version because of one or more facts received from the fact repository indicating the software version number. The one or more applicability rules 403 ~~407~~ may also check operating system version, platform/system version number, storage limits of the system, and software packages installed on the system. Other information may also be within the scope of the invention for the one or more applicability rules 403 ~~407~~ to check.

Please amend the paragraph on page 17 beginning on line 13 as follows:

If evaluating the one or more applicability rules 403 ~~407~~ returns a false, or some other negative identifier, the rest of the check including the one or more condition rules may not be evaluated. In another embodiment, a true or a positive identifier may indicate that the rest of the check does not need to be evaluated. Not evaluating the rest of the check may save evaluation time and eventually lead to faster product issue detection. In one embodiment, the one or more applicability rules in each check received by the knowledge automation engine may be evaluated before any of the one or more condition rules are evaluated. Also, in one embodiment, the check may not have applicability rules 403 ~~407~~.